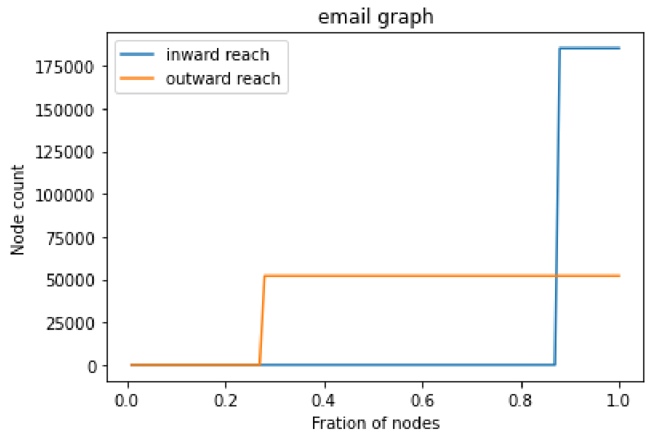
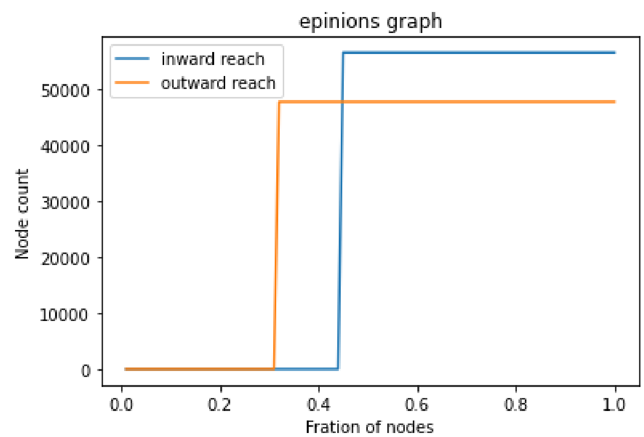
outward reach 2018: 52104
inward reach 2018: 1
outward reach 224: 47676
inward reach 224: 56459

These are the sizes of the sets of nodes that are reached using BFS method, which starts at the specified nodes. If a node reaches a large number of nodes using in-links/out-links, then the node is in OUT/IN, or SRC in special case. This is the same method described in Broder et al.

We can obviously see that node 2018 of the email graph is in SRC and node 224 of the epinions graph is in IN



We can estimate SRC to be comprised of less than 16% of the nodes, since it can at most have 50,000 nodes. Thus, IN and OUT can be estimated to be more than 50% and 3% of the nodes respectively.



Based on the fraction of modes that has low count in inward reach-ability and outward reach ability, we can cap the fraction of nodes in IN and OUT to around 40% and 30% respectively. Thus, SRC can be estimated to be in the range of 20% to 40%

The sizes of each components in both graphs are estimated using the same method as in Broder et al.,. We know that the size of SRC is the size of the biggest SCC. We know that SRC+IN/SRC+OUT is equal to the number of nodes that the BFS method can reach if it explodes using inward/outward links. TUBES+TENDRILS have nodes that are in the biggest WCC but not in SRC or IN or OUT. DISCONNECTED nodes are outside of the biggest WCC.

email graph
Total size: 265214
SRC size: 34203
Biggest WCC size: 224832
IN size: 140797
OUT size: 15797
TUBES+TENDRILS size: 34035
DISCONNECTED size: 40382

epinions graph
Total size: 75879
SRC size: 32223
Biggest WCC size: 75877
IN size: 22777
OUT size: 17777
TUBES+TENDRILS size: 3100
DISCONNECTED size: 2

email graph connection rate: 0.159

epinions graph connection rate: 0.456

These fractions are from 1000 sampled pairs of each graph. We would expect this to convert to the fraction $\frac{IN+SRC}{IN+SRC+OUT}\frac{OUT+SRC}{IN+SRC+OUT}$

1. Since the reset distribution is uniform, we can calculate:
   $V_{4,5} = V_{1,4,5} - V_1$
   $V_3 = V_{3,4,5} - V_{4,5}$
   $V_2 = V_{1,2,3} - V_1 - V_3$ Assuming the vectors are adjusted accordingly to reset rates.

2. we can't since the best we can do is calculate $\{4, 5\}$

3. $V_1 * 0.1 + V_3 * 0.3 + V_2 * 0.2 + V_{4,5} * 0.2$

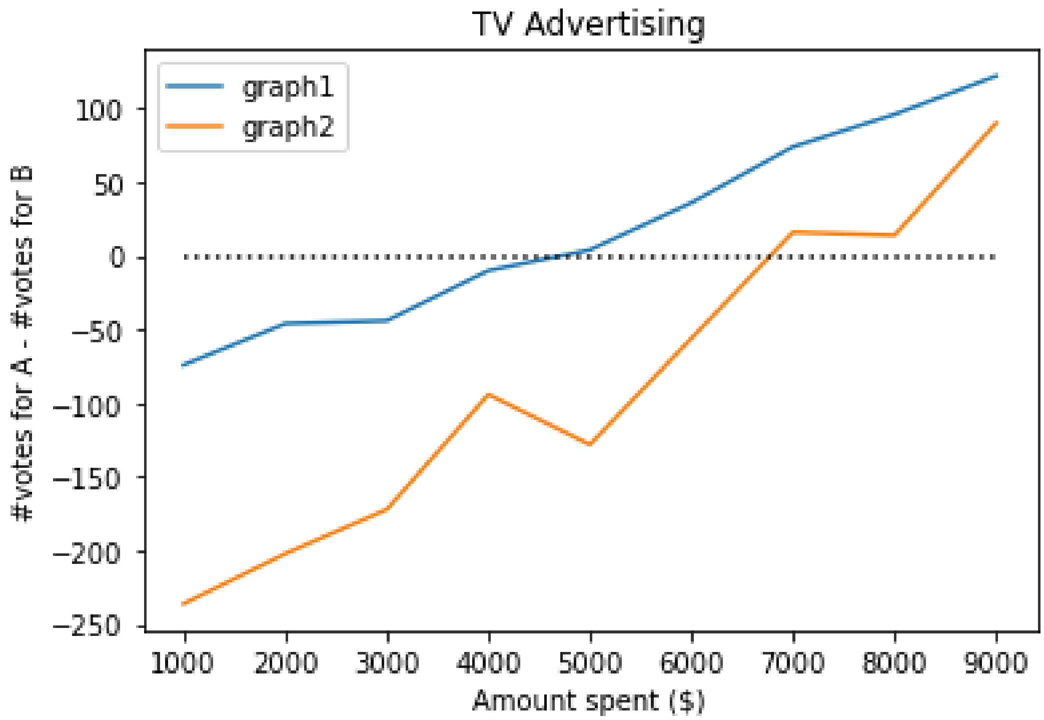
The set of computable personalized vectors is: $\{V_r^p | V_r^p$ is a linear combination of all computed $V_r\}$. V_r^p is the personalized vector with reset probabilities p and reset points r.

We can easily see that a personalized vector can be calculated using linear combinations of the vectors that have been computed (refer to question 2.3 for easier visualization)

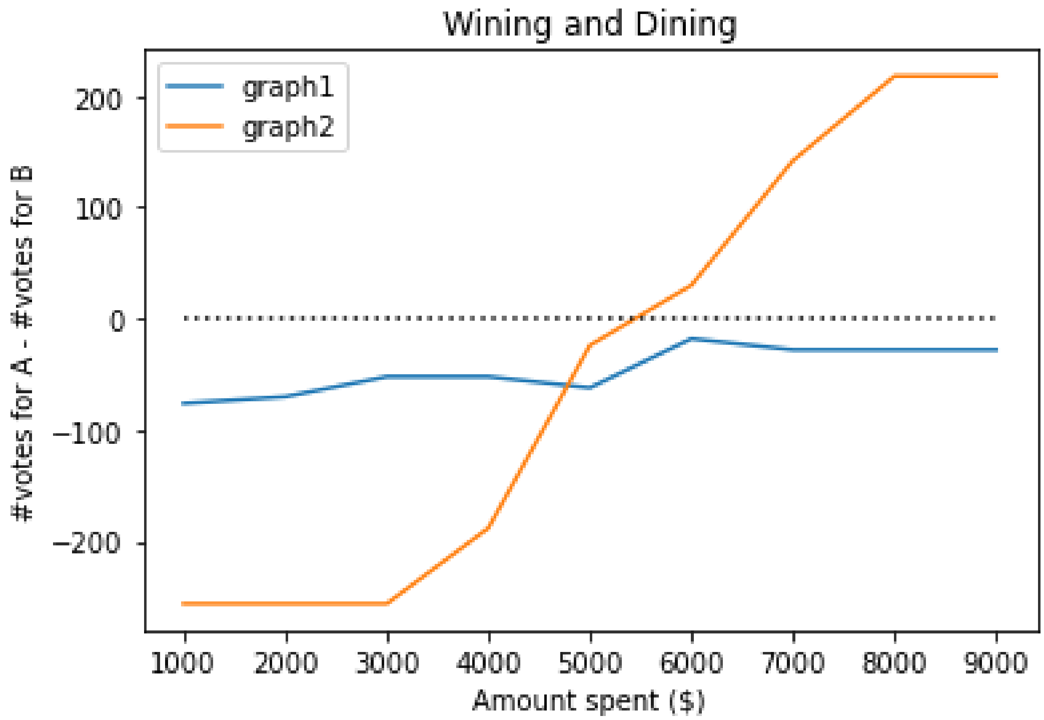
$$
\begin{aligned}
&r = \mathbf{A}r \\
\Rightarrow &r = [\beta \mathbf{M} + \frac{1-\beta}{N}\mathbf{1}\mathbf{1}^T]r \\
\Rightarrow &r = \beta \mathbf{M}r + \frac{1-\beta}{N}\mathbf{1} \; Q.E.D
\end{aligned}
$$

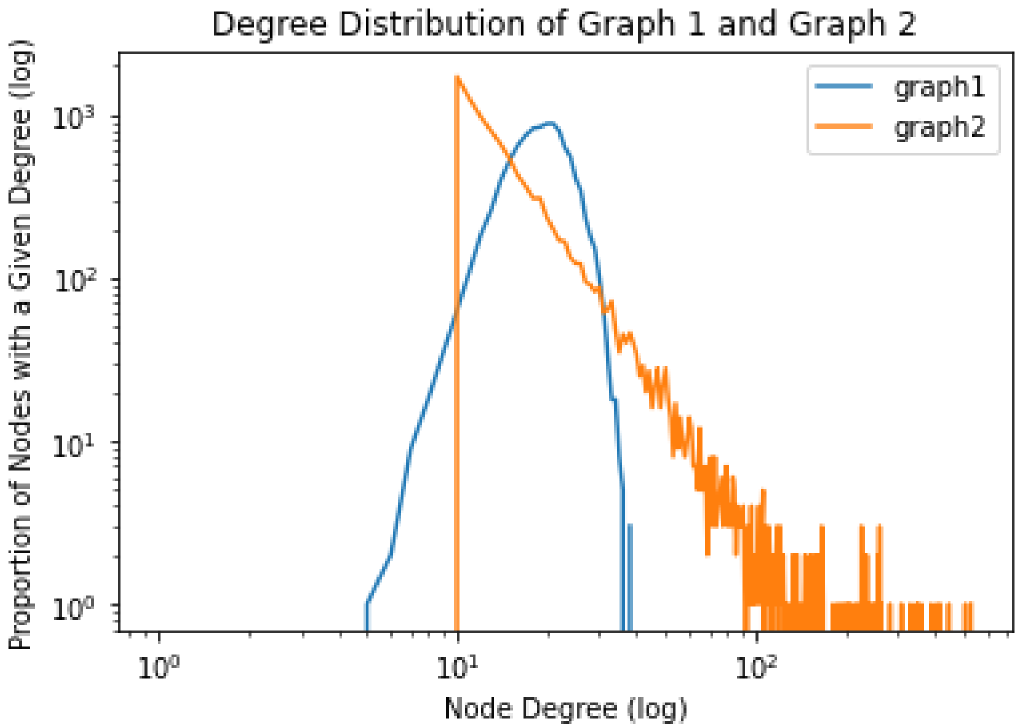
In graph 1, candidate B wins by 96 votes
In graph 2, candidate B wins by 256 votes

On graph 1, the minimum amount you can spend to win is 5000
On graph 2, the minimum amount you can spend to win is 7000

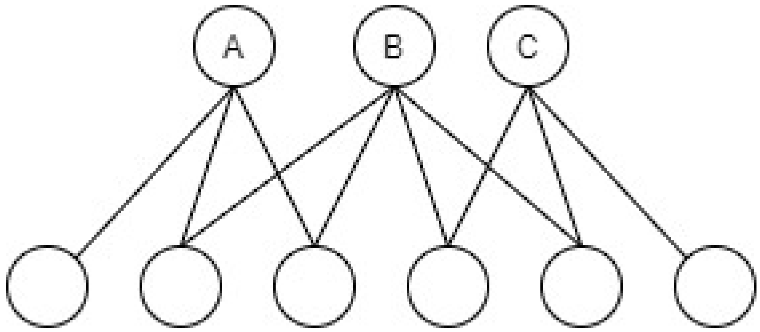


Q3: On graph 1, the minimum amount you can spend to win is None
On graph 2, the minimum amount you can spend to win is 6000





We can obviously see from the graph that the edges in graph 2 are more evenly distributed. Therefore, the high-roller effect is less pronounced in graph 2

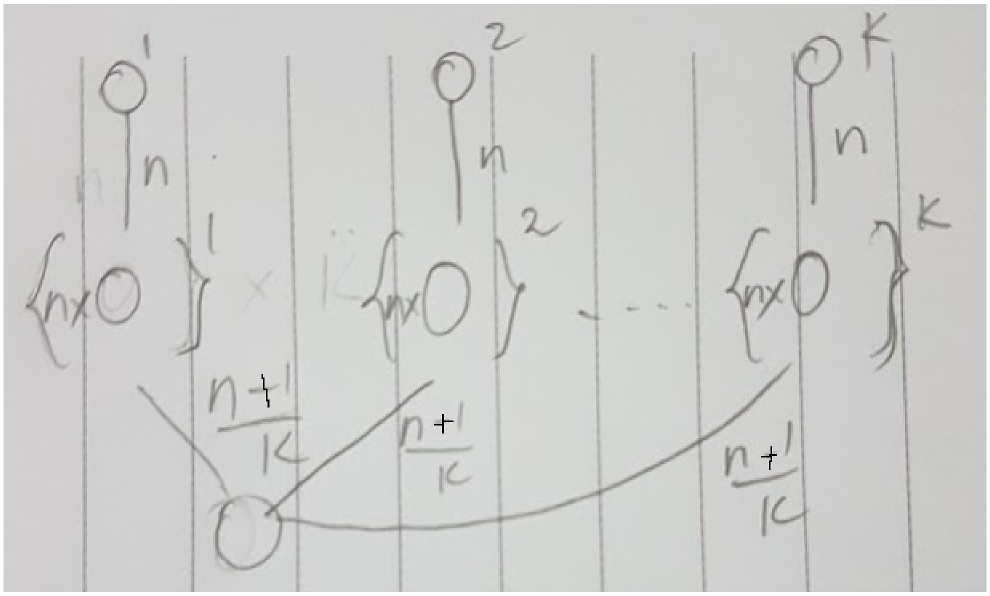


The greedy algorithm can only cover 7 nodes at best. However, the optimal solution for k=2 is 8 with S = $\{A, C\}$.

Following the graph described in 4.4, we can see that if $n \geq 14$ then we can get $f(S_i)$ to be at most 0.8 the value of $f(T)$ (Welcoming all better answers. I could not think of a cleaner one and im not drawing this one out lol)

1. $f(S_i) = f(S_{i-1} \cup \{u\}) = f(S_{i-1}) + f(n)$
2. $\|X_u\| = 1$

Any one of these properties is sufficient.



We can make n arbitrary big to arbitrary extend δ_i. One example is 4.2 where n=3 and k=2.

Information sheet
CS224W: Machine Learning with Graphs

Assignment Submission Fill in and include this information sheet with each of your assignments. This page should be the last page of your submission. Assignments are due at 11:59pm and are always due on a Thursday. All students (SCPD and non-SCPD) must submit their homework via GradeScope (http://www.gradescope.com). Students can typeset or scan their homework. Make sure that you answer each (sub-)question on a separate page. That is, one answer per page regardless of the answer length. Students also need to upload their code on Gradescope. Make sure to upload all of your code as .py files.

Late Homework Policy Each student will have a total of *two* late periods. *Homework are due on Thursdays at 11:59pm PT and one late period expires on the following Monday at 11:59pm PT*. Only one late period may be used for an assignment. Any homework received after 11:59pm PT on the Monday following the homework due date will receive no credit. Once these late periods are exhausted, any assignments turned in late will receive no credit.

Honor Code We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down their solutions independently, i.e., each student must understand the solution well enough in order to reconstruct it by him/herself. Students should clearly mention the names of all the other students who were part of their discussion group. Using code or solutions obtained from the web (GitHub/Google/previous year's solutions etc.) is considered an honor code violation. We check all the submissions for plagiarism. We take the honor code very seriously and expect students to do the same.

Your name: ____

Email: ______________________________ **SUID:** ________________________

Discussion Group: ____

I acknowledge and accept the Honor Code.

(Signed) ____